SCALABLE DATABASE MANAGEMENT SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to database systems, and more particularly to a scalable, network-based database system.

2. Description of the Related Art

Conventional database systems now provide Internet interfaces that allow multiple users to access databases through the Internet. Particularly, a first registered user accesses and modifies values in a database through their web browser. A second registered user similarly accesses the database through their browser. This allows the different users to access the database from separate locales through a regular network (e.g., Internet) connection.

An example of a conventional database system implements a web server that interfaces with web clients, and an application server that receives and sequentially processes received database requests by accessing a back end relational database server. There are several problems with this architecture. One is that the sequential processing of database requests allows a single time consuming or problematic database request to delay the processing of subsequent requests that could have been easily handled. Particularly in systems where many different groups of users access many separate databases, the application server becomes a significant processing bottleneck.

Another problem with these systems is that a majority of the database users may commonly reside in a particular location relative to the application server. Inefficiencies arise when numerous remote users access large amounts of data traversing many network segments.

Still another problem with these systems is that it is difficult to increase capacity. Merely adding application servers would not solve the problems of these systems. Indeed, such may create additional problems, as this merely pushes back the bottleneck to a router which must distribute requests across multiple application servers, and may also require a lock server to manage concurrent access.

Thus, there remains a need for a database system that handles database requests more efficiently, removes bottlenecks associated with conventional systems, facilitates system growth without sacrificing database integrity or efficiency, and allocates database servers to efficiently serve their clients.

SUMMARY OF THE INVENTION

The present invention allows the management of numerous databases that are accessed by users from various remote locations, allows capacity to be increased in a scalable fashion, manages and allocates database requests and failures more efficiently, and facilitates migration of database assignments.

In one aspect, the present invention provides a number of web based request handler modules (RHM) through which clients interface with the database management system. These request handler modules communicate with a master control module (MCM) that manages the assignment of numerous database application modules (DBAM) to databases. Each DBAM can communicate with a file server or the like that persistently stores the databases managed by the system, in centralized or decentralized fashion. The DBAM can then load assigned databases from the file server, for handling requests, and can unload databases when their assignment terminates.

When an RHM receives a database request from a client, it sends identifying information for the relevant database to the MCM. The MCM determines an assigned DBAM for the database, and returns location information for the assigned DBAM to the RHM. The RHM uses this location information to connect with and send the database request to the assigned DBAM for handling. The assigned DBAM handles the database request and returns its results to the RHM for provision to the client.

These features facilitate deployment of multiple web based request handlers that interface with various users dispersed throughout a network, such as the Internet. The web based request handlers do not need to be modified and need not be concerned about the introduction of additional database servers to handle increasing system loads, as the master control module maintains the association of databases to database servers. Further, by identifying databases and mapping them to DBAMs by the MCM, the system prevents request handlers from sending database requests to the incorrect database server, and prevents multiple request handlers from concurrently requesting separate database servers to perform database operations on the same database, retaining database integrity.

In one embodiment, the MCM maintains a mapping of unique database identifiers to currently assigned DBAMs. When the RHM receives a database request from a client, it extracts the database identifier from the request and sends it to the MCM, which examines the mapping to locate the currently assigned DBAM for that identifier and thus the database corresponding to the identifier. The mapping may be variously implemented but in one example the database identifiers are mapped to the IP address and port number corresponding to the currently assigned DBAM. This information is returned to the RHM, which uses it to send the database request to the appropriate DBAM. The DBAM may already own the database and have it loaded upon receipt of the request from the RHM. The DBAM can also communicate with the file server to load the database corresponding to the DBID. The database can be accessed by file name, and a shared storage can list the association of file names to DBIDs.

Requests from the clients to create databases are also fulfilled. In one example, the RHM receives such a request and identifies the desired action to the MCM, which generates a new DBID, assigns an available DBAM to the request, and gives the DBAM location information to the RHM. The RHM (or MCM) can then use the location information to send the request to create a new database to the DBAM, which creates the new database (e.g., by receiving user input that defines the fields in the database, or by copying a database template, etc.). The DBAM then retains the created database in memory to handle database requests. The new DBID can then be used to appropriately direct future database requests for the created database to the assigned DBAM.

The MCM can also assign databases to alternative DBAMs, for any number of reasons. In one example, improved failure recovery is provided, as the MCM coherently and efficiently reassigns a database to an alternative DBAM when a currently assigned DBAM fails or is non-responsive. This is accomplished by retaining a list of available DBAMs, terminating the assignment of a database to a current DBAM, identifying an alternative DBAM from the list, sending the location information for the alternative